System Test Specification

STORC DASHBOARD PROJECT

CD Jam

CSC 191 – FALL 2015 | 11/18/2015

Table of Contents

1. INTRODUCTION

This is the Software Test Specification document for the STORC Dashboard Project sponsored by Dr. Michael Christensen.

This project is being undertaken by the CD Jam software development team. The team is comprised of undergraduate students majoring in Computer Science at California State University, Sacramento. The team members are enrolled in a two-semester senior project course required of all undergraduate majors. Successful delivery of the desired software product will fulfill the senior project requirement for the student team members.

PROJECT SPONSOR:
Name: Michael Christensen
Title: Assistant Vice President for Risk Management Services & Director of STORC
Organization Name: Sustainability Technology Optimization Research Center (STORC)
Contact Information:
Phone: (916) 278-5252
Email: storc@csus.edu

CD JAM DEVELOPMENT TEAM:
Name: Cole Culler
Contact Information:
Phone: (530) 575-7683
Email: cbroski.culler@gmail.com
Name: David Grapentine
Contact Information:
Phone: (707) 471-8749
Email: davidjoaograpentine@gmail.com
Name: Ashley Gregory
Contact Information:
Phone: (209) 304-1884
Email: aa5gregory@gmail.com
Name: John Jones
Contact Information:
Phone: (916) 475-8460
Email: felixequal@gmail.com
Name: Michael Smith
Contact Information:
Phone: (916) 842-8339
Email: infinitlyill@gmail.com

1.1 Purpose

The purpose of the STS is to 1) describe the plan for testing the software, and 2) specify the test cases and test procedures necessary to demonstrate that the software satisfies the requirements as specified in the project's System Requirements Specification document.

1.2 Scope

The plan contains a list and brief description of the use cases to be tested and the software components associated with each test case. The plan also provides a schedule for the testing and the assignment of team members to their respective testing tasks. The process for documenting resolving software errors and/or anomalies that are found during the testing is also specified. The test specification includes a list of the features to be tested for each of the use cases, the description each test case needed to fully test the use case, and the test procedures, or steps, necessary to execute each of the test cases.

1.3 Definitions, Acronyms, and Abbreviations

1.3.1 Definitions

Administrator: A super user that oversees other users and the projects within STORC.

Aquaponics: A cycle between hydroponically grown plants and aquatic animals, in which the waste produced from animals supplies nutrients for plants which in turn purifies the water.

Biodiesel: A substitute for diesel created by a biological chemical reaction.

Dashboard: A collection of data laid out in an easy to read format represented in a graphical format.

Photovoltaic Cell: A device that delivers an electric current as a result of a chemical reaction from the rays of the sun.

Principal Investigator: A user that in charge one or more STORC projects and oversees one or more STORC Technicians. Principal Investigators are most often faculty or staff members at CSUS.

Public: Any user that is not directly involved with STORC or STORC activities.

Technician: A user who works on a project overseen by a PI. This user generally monitors and collects data directly from one or more project stations. These are usually student volunteers.

Vermiculture: The cultivation of worms used for composting materials.

Webmaster: This user is in charge of maintaining and updating the default web page and widgets for the STORC Dashboard Project.

1.3.2 Acronyms

CSc – Computer Science
CSUS – California State University, Sacramento
ECS – College of Engineering and Computer Science
ERD – Entity Relationship Diagram
GUI – Graphical User Interface
HTML – Hyper Text Markup Language
IRT – Information Resources and Technology
IT – Information Technology
MySQL – My Structured Query Language
PI – Principal Investigator
PMP – Software Project Management Plan

SDS – Software Design Specification
SRS – Software Requirements
STR – System Test Report
STORC – Sustainability Technology Optimization Research Center
STS - Software Test Specification
UM – User Manual
UML – Unified Modeling Language
WCM – Web Content Management

1.3.3 Abbreviations

Admin – Administrator
CSc 190: Computer Science Senior Project - Part 1
CSc 191: Computer Science Senior Project - Part 2
Tech – Technician

1.4 References

Buckley, Bob. *CSc 190-01 Senior Project: Part 1.* CSUS, Dec. 2014. Web. 22 February 2015. http://athena.ecs.csus.edu/~buckley/CSc190/CSc190.html

STORC. CSUS STORC. n.p. Web. 22 February 2015. http://www.csus.edu/storc/about.html

1.5 Overview of Contents of Document

2. TEST PLAN DESCRIPTION this section provides a summary of the Use Cases and the plan for carrying out the system test phase of the team's software development process. More specifically, this section contains a brief description of each Use Cases to be tested, the team member (or members) assigned to test each Use Case, the testing schedule, and the risk management plan. *3. TEST DESIGN SPECIFICATION* this section describes the details of the test approach, lists the use cases that are and are not to be tested, lists the environmental needs, and details the pass/fail and suspension/resumption criteria. *4. TEST SPECIFICATION* this section contains subsections for each of the FEATURES to be tested. Each subsection specifies the USE CASES to be tested, the procedures necessary to run the test cases, the items being tested. *5. SYSTEM TEST / REQUIREMENTS TRACEABILITY* this section provides for a cross referencing of each test case to its software requirement specification (or specifications) and also to its design component (or components). The appropriate section and its title in each document are provided. 6. APPROVALS this section contains the list of the key signatures necessary to sign-off on the STS, thereby agreeing to the scope and content of the test plan and test cases specified within the document. Approval constitutes a guarantee that the development team has produced a test specification sufficient for validating the software to be delivered to the sponsor.

2. TEST PLAN DESCRIPTION

2.1 Product Summary

The STORC Dashboard Project is an application that will allow employees and the public to see the status of STORC's projects and track sustainability over time. The dashboard will be able to collect data through the sensors associated with each project as well as allow students to manually enter data. The data collected will be displayed to both students and the public through several graphical user interfaces or widgets. Each individual employee will be able to customize their own view allowing each user to view information that pertains to their current project within STORC. This will also provide an interface to add new projects and integrate them within the database and the dashboard. The STORC Dashboard Project will be a dynamic application instead of a static application.

The following table contains a listing of the use cases; the system's features associated with each use case along with its files and database tables.

	Feature	Use Cases	Components
1	Homepage Management Interface	UC4. Customize Dashboard	Dynamic widgets, database tables of projects and data
2	Project Selection Interface	UC4. Customize Dashboard	Database tables associated with the project
3	Data Input Interface	UC3. Input Data	Database tables associated with the project to put data into
4	Review and Submit Data Interface	UC1. Manage Data UC3. Input data	Database tables associated with the project
5	Add/Remove/Edit Users Interface	UC2. Manage Users	SacLink Username and List of Projects
6	View/Select Pending Data	UC1. Manage Data	Database tables associated with that project
7	Select/Edit Project	UC1: Manage Data UC4. Customize Dashboard	Database tables associated with the project

2.2 Responsibilities

The following table contains a list of each team member and their assigned use case along with the set-up required for testing each use case.

Use Case	Team Member	Set-up
UC1: Manage Data	Ashley	Database Setup
UC2: Manage Users	John	SacLink IRT Connection
UC3: Input Data	Cole	Database Setup
UC4: Customize Dashboard	David	Database Setup
UC5: Login	Michael	SacLink Login (provided by IRT)

2.3 Schedule

Feature	Start Testing	Resolve Bugs	Finish Testing
Homepage Management Interface	11/24/2015	12/4/2015	12/7/2015
Project Selection Interface	11/24/2015	12/4/2015	12/7/2015
Data Input Interface	11/1/2015	11/27/2015	12/7/2015
Review and Submit Data Interface	11/10/2015	11/30/2015	12/7/2015
Add/Remove/Edit Users Interface	11/10/2015	11/30/2015	12/7/2015
View/Select Pending Data	11/10/2015	11/30/2015	12/7/2015
Select/Edit Project	11/10/2015	11/30/2015	12/7/2015

3. TEST DESIGN SPECIFICATION

3.1 Testing Approach

The information below lists how CD Jam will approach testing:

- Conversion Testing
- Interface Testing
- Security Testing
- Recovery Testing
- Performance Testing
- Regression Testing
- Constraints

Tests will be performed by entering data into the various web pages and observing the output data. In most cases validation will require database queries to ensure that the proper data was created, retrieved, updated or deleted.

3.2 Feature or Combination of Features Not to be Tested

Some features of the project that will not be tested include:

- **Look and Feel** - The overall look and feel of the project.
- **Sensor Compatibility with the STORC Server -** If sensors are unable to report data to the server over a period CD Jam will not be responsible for testing or correcting the issue.
- **Login Reliability (handled by IRT)** - All login details will be handled by IRT with no input from CD Jam.
- **STORC Webpage Reliability (webpage maintained by IRT) -** We are not responsible testing csus.edu releated issues. If there is an issue with the domain or website, IRT will be responsible for correcting the issue.

- **Screen Resolution Inconsistency -** Depending on the individual user's resolution settings, the software's interface may look different then what is expected.
- **Network Connection -** STORC has limited network connectivity and throughout the facility the Wi-Fi signal is unreliable. Any LAN connections issues with the STORC computer will handled with IRT. If the network is lost during use of the program, CD Jam will not be responsible for any lost data and no testing will be performed to cover lost connections.
- **Unanticipated Error Conditions -** It is possible bugs/issues may arise that are outside of our test cases.

3.3 Environmental Needs

To successfully test the STORC Dashboard Project, CD Jam requires an IRT Server for login as well as the STORC Server and a Database, test database or STORC database, to be in place and functioning. All testing will be done using Microsoft Visual Studio and/or a web browser. A test database is located on CD Jam's ECS project account that the team has received from ECS IT. No specific testing will be used to test the STORC Dashboard Project. There are no hardware requirements needed since the project is performed solely on the tester's web browser or within our coding IDE, Microsoft Visual Studio. Once testing is complete IRT will host the STORC Server and STORC Database which will be located on a Virtual Machine.

3.4 Suspension / Resumption Criteria

All tests will be run to the fullest extent until either complete or the tester runs into a bug. If the tester/testers are able to complete all testing requirements for that test case with no issues occurring than CD Jam will consider the following feature done after another individual has verified the tester's results.

When a bug is found during testing the tester/testers will stop all testing on that feature and fill out a report about the bug providing all details about the bug. The CD Jam team will then be notified as soon as possible. It is then the responsibility of the lead coder of that feature to fix the bug and ensure the bug does not happen again. Once the lead coder has fixed the issue testing will be started from the beginning and continue until complete or another bug is found.

Once all components of the project are considered to have passed the testing stage, all components will be tested together as a whole to ensure the project functions cohesively. If a problem occurs at this stage of testing the lead coder in charge of the malfunctioning section will fix the issue and case testing for the whole project will restart from the beginning.

3.5 Risk and Contingencies

STORC will provide all necessary resources to create the environment specified in the SRS. If access to this environment is not available for testing purposes CD Jam will create a similar environment in order to test the STORC Dashboard Project.

If the testing schedule is significantly impacted by system failure, CD Jam will spend as much time as possible on correcting and documenting the issue(s). If a specific component of the project is causing major issues the team will take the component out of the project and make the necessary changes until the component has been fixed. The team will then reintegrate the component back into the project and the component will then be reevaluated. If that does not fix the problem this issue will be brought up with our sponsor and a discussion will take place in order to evaluate the necessity of the component.

If a team member is unable to complete coding, testing, or debugging within an acceptable amount of time the individual will be talked to by the team as a whole and by the team lead. If this continues the CSC 191 professor, Dr. Salem will have a discussion with the individual. The other members of the team will then assume the responsibilities, including testing, of the non-functioning team member. This team member will receive a lower grade in the class.

4. TEST SPECIFICATION

4.1 Test Procedures

1. The main coder will create a document explaining how the feature should behave.
2. Once the document has been created the main coder will alert the group that the feature is ready for testing and commit the code to GitHub.
3. A team member will then be assigned to test the feature and retrieve the code from GitHub.
4. The main tester will follow the instructions, starting in Section 4.4, to test the feature.
    a. If the main tester finds any issues they are required to fill out a Software Problem Report (SPR) and upload the report to CD Jam’s Google Drive under the folder “SPR”.
    b. The tester must then notify the main coder that a problem has occurred and a SPR has been filled out for that feature.
    c. It is the responsibility of the main coder to correct this issue.
    d. The main coder will then start the test procedures over again.
5. If all test cases for that feature are fulfilled the tester will notify the team that no issues have been found in that feature and it can be committed to the master GitHub branch.

   **Note: If testing for a feature is interrupted, we require that the test is restarted from the beginning.

4.2 Test Procedures Conventions

1. Installation of Software Instructions:
    a. Tester will navigate to GitHub and sync their account.
    b. They will then open the specific feature to be tested in Visual Studio 2015.
    c. The application will then be ran from the Visual Studio Debugging GUI.
2. For each test case the tester will start from the homepage and navigate to the feature being tested.

3. The tester will follow the test cases specified and enter the data selected from the pool of possible infinite inputs.
4. The tester will then verify the inputs result in reasonable outputs, which can be found within the database using queries.

4.3 Test Data

Feature 1: Customizing Widget

Customizing a widget does not require inserting or deleting data, however it does require modifying the visual output of the data, and may also include changing the data source; a different column or calculation may be used. Data in the database to test the change in data source will need to be pre-loaded to test this case. We will use the food waste table to test widget customization.

Data will initially be in a csv file, which will be inserted into the the foodWasteDataPoint table using the following SQL command:

```
BULK INSERT CDJAM.dbo.foodWasteDataPoint
        FROM 'C:\foodwaste.csv'
        WITH
        ( FIELDTERMINATOR = ',', ROWTERMINATOR = '\n' )
```

After the bulk insert, foodWasteDataPoint data columns in the table should appear as below:

ID	Date	Weight	Grains	Fruit	Veg	Dairy	Paper	Coffee
1	2015-01-02	1	5	10	12	1	10	5
2	2015-01-03	2	6	7	4	2	15	15
3	2015-01-04	4	6	9	12	1	20	17
4	2015-01-05	10	2	3	4	3	21	23
5	2015-01-06	8	3	6	13	4	11	10
6	2015-01-07	7	2	3	5	6	17	7

Test Case 1: Deleting a Widget

The database must have a test user in the Users table so that deletion of a widget will be reflected in the user's profile configuration column.

Insert a test user into the database with the following command:

```
INSERT INTO Users
VALUES ('test', 'firstname','lastname', 'admin',,);
```

The Users directory should look like the following:

UserID	Username	FirstName	LastName	UserType	ConfigFile
1	Test	Firstname	Lastname	Admin	

After deleting a widget, the default configuration file that is generated will reflect the change in the default widget view, i.e the deleted widget will no longer show on the user's homepage. The configuration file should be viewed to verify this.

Test Case 2: Adding a Widget

See Test 2: deleting a widget. The user table should be cleared and reinitialized with the data from Test 2. Use the following command to delete the single line in the User table:

```
DELETE FROM Users
WHERE ID = 1;
```

Test Case 3: Moving/Resizing a Widget

Data: Resizing and moving a widget does not require any other changes besides adding configuration details (widget position/size) to the user's configuration file in the Users table.

Procedure: Set up the database the same way as both tests 2 and 3.

Feature 2: Create New Project

Data: No pre-condition for project table required. Post-condition: Projects Table should appear as below:

ID	Name	Location	DataOnline	DomainID
1	FoodWasteCollection	Section 4	2015-11-10	Collection

Feature 3: Data Input Interface

Test Case 1: Input Data

Precondition: Re-populate the FoodWasteDataPoint as in Feature 1: Test Case 1 using the following SQL command:

```
BULK INSERT CDJAM.dbo.foodWasteDataPoint
FROM 'C:\foodwaste.csv'
WITH
( FIELDTERMINATOR = ',', ROWTERMINATOR = '\n' )
```

Post-Condition: None. Data is not entered until reviewed and submitted by an administrator.

Feature 4: Review and Submit Data Interface

Test Case 1: Review and Submit Data

Pre-Condition: Table data should appear as in feature 1 - test case 1

Post-Condition: After inserting data into the FoodWasteDataPoint table using the user review and submit interface the table data should appear as below. Inspect the database table to verify.

ID	Date	Weight	Grains	Fruit	Veg	Dairy	Paper	Coffee
1	2015-01-02	1	5	10	12	1	10	5
2	2015-01-03	2	6	7	4	2	15	15
3	2015-01-04	4	6	9	12	1	20	17
4	2015-01-05	10	2	3	4	3	21	23
5	2015-01-06	8	3	6	13	4	11	10
6	2015-01-07	7	2	3	5	6	17	7

Feature 5: add/remove/edit user interfaces

Test case 1: Add User

Precondition: Table should be appear as in Feature 1 – Test Case 2:

UserID	Username	FirstName	LastName	UserType	ConfigFile
1	Test	Firstname	Lastname	Admin	

Procedure: Enter the following data through the add user interface:

Username: Student1
First Name: Stu
Last Name: Dent
User Type: Student
Post-Condition: User table should appear as below. Check the database to verify:

UserID	Username	FirstName	LastName	UserType	ConfigFile
1	test	Firstname	Lastname	Admin	
2	Student1	Stu	Dent	Student	

Test Case 2: Edit User Permissions

Pre-Condition: User table should appear as below, identical to previous test case:
Procedure: Use the edit user interface to elevate the Student1 user to administrator.
Post-Condition: The User table should appear as below. Check database to verify:

UserID	Username	FirstName	LastName	UserType	ConfigFile
1	test	Firstname	Lastname	Admin	
2	Student1	Stu	Dent	Admin	

Test Case 3: Remove User
Pre-Condition: User Table should appear as below from the Feature 5 – Test Case 2:

UserID	Username	FirstName	LastName	UserType	ConfigFile
1	test	Firstname	Lastname	Admin	
2	Student1	Stu	Dent	Admin	

Procedure: Remove user via the user interface.
Post-Condition: User table should appear as below. Check database to verify:

UserID	Username	FirstName	LastName	UserType	ConfigFile
1	test	Firstname	Lastname	Admin	

4.4 Feature 1: Homepage Management Interface

4.4.1 Test Case 1: Customizing Widget

1. Starting on the STORC Dashboard Homepage click on the "Edit Widget" button
2. This will bring up the "Edit Widget" View
3. From there the user will be able to Customize the look of the Widget on the Dashboard
    a. Users will be able to customize the following
        i. Widget "Graphs, Calendar's, Pie Charts, etc." type
        ii. Data source
        iii. Color scheme of the widget
4. The user can then save or discard the changes made by clicking on the "Save" or "Cancel" buttons

4.4.1.1 Test Items

Feature	Description	Pages
Homepage Management Interface	Customize Widget - Right clicking on one of the widgets, the user will be presented with menus and dropdowns that can change the widget's data source(s), widget's chart type, colors, and other relevant settings.	EditWid.html EditWid.cs EditWid.js EditWid.aspx

4.4.1.2 Input Specifications

1. Right click on a widget
2. A menu will appear giving the user the choice of changing the following
    a. Type of graph
        i. Bar Graph
        ii. Pie Chart
        iii. Line Graph
    b. Color of widget
    c. Data source
        i. Each data source will be tied to specific data collected for that project

4.4.1.3 Output Specifications

1. The STORC Dashboard Homepage will refresh and display the changes the user made.
2. This layout will be saved to the database in order to bring up the same layout when the user logs back into their dashboard.

4.4.1.4 Intercase Dependencies

1. The user must login successfully and needs to be an active user.

4.4.2 Test Case 2: Deleting Widget

1. Starting with the STORC Dashboard Homepage click on the "Edit Widget" button
2. This will bring up the "Edit Widget" View
3. From there the user will be able to delete the Widget on the Dashboard
4. The user can then save or discard the changes made by clicking on the "Save" or "Cancel" buttons

4.4.2.1 Test Items

Feature	Description	Pages
Homepage Management Interface	Delete Widget - Clicking the X-icon in the top right corner of the widget, the widget can be deleted. There will be a prompt to verify the user's decision before this widget deletion.	DeleteWid.html DeleteWid.cs DeleteWid.js DeleteWid.aspx

4.4.2.2 Input Specifications

1. Click the "X" on the top right hand corner of the widget

4.4.2.3 Output Specification

1. The STORC Dashboard will refresh and display the changes that the user applied.
2. This layout will be saved to the database and when the user logs out it will reload their dashboard.

4.4.2.4 Intercase Dependencies

1. The user must login successfully and needs to be an active user.

4.4.3 Test Case 3: Adding a Widget

1. Starting on the STORC Dashboard Homepage click on the “Edit Widget” button
2. This will bring up the “Edit Widget” View
3. From there the user can click on the “Add Widget” button.
4. The user will then be able to add a widget of their choosing.
5. The user can then save or discard the changes made by clicking on the “Save” or “Cancel” buttons

4.4.3.1 Test Items

Feature	Description	Pages
Homepage Management Interface	Add New Widget - By clicking the Add New Widget button, the user will be prompted for widget data sources, chart types, colors, and other relevant settings. After configuring the new widget’s settings, a new widget will be added to the dashboard where the user can position and resize as they see fit.	EditWid.html EditWid.cs EditWid.js EditWid.aspx

4.4.3.2 Input Specifications

1. Click on the “+” button on the top right hand corner of the screen.
2. After choosing the desired widget, the user can click on either the “Save” button or the “Cancel” button.
   a) If the user selects the “Save” button then the newly added widget will be saved for that specific user.
   b) If the user selects the “Cancel” button then the newly added widget will not be saved for that specific user.

4.4.3.3 Output Specifications

1. A window will appear prompting the user to add a specific widget
2. The user will be able to do the following when adding a widget:
   a. Select which type of widget to add
   b. Select the data source for the desired widget
   c. Select the color for the widget
   d. Customize the size and location of the widget
3. The newly added widget will be saved in the user’s profile.

4.4.3.4 Intercase Dependencies

1. The user must login successfully and needs to be an active user.

4.4.4 Test Case 4: Move/Resize Widget

1. Starting on the STORC Dashboard Homepage click on the "Edit Widget" button
2. Start by selecting the bottom right corner of the widget
   a. By selecting the bottom right corner of the widget, the user can:
      i. Drag the widget around the dashboard in order to move the widget to another location.
      ii. Resize the widget to a desired height and width.
3. Select the "Save" button on the dashboard screen in order to save any changes to the widget.

4.4.4.1 Test Items

Feature	Description	Pages
Homepage Management Interface	Move/Resize Widget - Moving and resizing a widget is as simple as selecting the upper right corner of the widget and dragging and panning to resize and situate the widget on the dashboard. It is likely that automatic snap functionality will be included to make this supporting feature more user-friendly.	EditWid.html EditWid.cs EditWid.js EditWid.aspx

4.4.4.2 Input Specifications

1. Click "Edit Widget" button on Dashboard page.
2. Then hover arrow over the bottom right corner of the widget that the user wants to edit.
   a. By keeping the arrow directly on the bottom right corner of the widget, the user will be able to move the widget around the dashboard.
   b. If the user moves the arrow close to the North, West, South, or East border of the widget then they will be able to resize the widget either vertically or horizontally.
3. After the user has moved and/or resized the desired widget, they then can click on the "Save" button in order to save all changes made.

4.4.4.3 Output Specifications

1. After the user hovers the arrow over the top right corner of the widget, the user will notice the arrow change to a specific symbol.
   a. If the arrow changes to 4 arrows pointing out, then that means the user is able to move the widget.
   b. If the arrow changes to two arrows pointing in opposite directions, then that means that the user can resize the widget.
      i. If the arrows are pointing in opposite directions from left to right, then the user can resize the widget horizontally.

ii. If the arrows are pointing in opposite directions from top to bottom, then the user can resize the widget vertically.
iii. If the arrows are pointing in opposite directions at a 45 degree angle, then that means the widget can be resized both vertically and horizontally at the same time.

2. A save button will be visible once any changes are made to the widget.
3. All widgets will be saved for that specific user profile.

4.4.4.4 Intercase Dependencies

1. The user must login successfully and needs to be an active user.

4.4.5 Test Case 5: Create New Project

1. Starting on the STORC Dashboard Homepage click on the “Edit Widget” button.
2. At the top of the Dashboard, click on the tab labeled, “Create a New Project”.
3. The user will be taken to a new page where they will be able to do the following:
    a. Choose the station(s) necessary for the project.
    b. Create specific widgets for the project:
        i. When creating the widgets for the project, the user will need to choose the data sources necessary for the widget.
        ii. They can also choose the color, size and location here as well.
4. The user can then save or discard the changes made by clicking on the “Save” or “Cancel” buttons at the bottom of the screen.

4.4.5.1 Test Items

Feature	Description	Pages
Homepage Management Interface	Create New Project- After clicking on the “Edit” button within the Dashboard, the user will see a “Create a New Project” tab at the top of the Dashboard. By clicking on this tab, the user will be taken to a new screen where they are prompted to create a new project and add it to their dashboard.	ManageData.html ManageData.cs ManageData.js ManageData.aspx ManageUsers.html ManageUsers.cs ManageUsers.js ManageUsers.aspx

4.4.5.2 Input Specifications

1. Click on the “Edit” button at the top of the STORC Dashboard.
2. Click on the tab labeled “Create New Project”.
3. Fill in all inputs necessary to create a new project.
    a. This may also include adding and customizing widgets that are to be used for the new project which will be done at a later time.

4.4.5.3 Output Specifications

1. Once the project has been created a review page will display on the screen.
2. The user's dashboard homepage will then refresh and will allow the user to create widgets based on the new created project.

4.4.5.4 Intercase Dependencies

1. The user must login successfully and needs to be an active user.

4.5 Feature 2: Project Selection Interface

4.5.1 Test Case 1: Select Project and Station

1. Starting with the STORC Dashboard Homepage click on the "Select Project" tab.
2. The user will be taken to a new page where they will be able to do the following:
    a. Choose a project and station(s) that requires manual input of data.
        i. Once a project is chosen the user can proceed to the input form by clicking "Input Data"
3. The user can then save or discard the changes made by clicking on the "Save" or "Cancel" buttons at the bottom of the screen.

4.5.1.1 Test Items

Feature	Description	Pages
Project Selection Interface	Select Project and Station - Users will select individual projects and stations that require manual input of data. Once selected they can proceed to the input form.	ManageData.html ManageData.cs ManageData.js ManageData.aspx

4.5.1.2 Input Specifications

1. Click the "Select Project" tab at the top of the STORC Dashboard
2. Choose a project from the list provided.

4.5.1.3 Output Specifications

1. Once the user selects a project they will be prompted to input data for the project selected.

4.5.1.4 Intercase Dependencies

1. The user must login successfully and needs to be an active user.
2. The user must have permissions to access specific data sources.

4.6 Feature 3: Data Input Interface

4.6.1 Test Case 1: Input Data

1. Starting on the STORC Dashboard Homepage click on the "Select Project" tab.
2. The user must then select a project by double clicking on the project.
3. The user will then be taken to a form tailored to the project selected.
4. The user must enter data into the appropriate fields.
5. After the user has completed the data input, they will then have to click on the "Submit" button in order to complete the data input process.

4.6.1.1 Test Items

Feature	Description	Pages
Data Input Interface	The user will be presented with a single row to fill out. This row will automatically show the column categories to make it easier to correlate a user's written data with the data to be entered. Once a row has been filled, a new row can automatically be generated to improve data entry flow.	InputData.html InputData.cs InputData.js InputData.aspx

4.6.1.2 Input Specifications

1. The user will be presented with an empty form.
   a. The user must fill out all the data that has a red asterisk next to the label.
   b. The user must also fill out the form fields with the correct data type.
2. Once the form is filled out, the user must select the "Submit" button in order to temporarily store the filled out form in the "Review Data Queue"
   a. After the data is submitted, it will be automatically sent to the Principal Investigator for approval.

4.6.1.3.Output Specifications

1. After entering the data, the user will be presented with a message confirming that the entered data has been sent to the Principal Investigator for approval.

4.6.1.4 Intercase Dependencies

1. The user must login successfully and needs to be an active user.
2. The user must have permissions to access specific data sources.
3. The user must have a dashboard setup for a specific project before entering new data.

4.7 Feature 4: Review and Submit Data Interface

4.7.1 Test Case 1: Review and Submit Data

1. After filling out the input form, the user will be able to review and edit the data before submittal to the PI/Admin.
    a. If changes need to be made to the data the user can do so here.

4.7.1.1 Test Items

Feature	Description	Pages
Review and Submit Data Interface	Once the user has finished filling out the input form, they will be able to review all data they have input and submit the data to the Principal Investigator for approval.	InputData.html InputData.cs InputData.js InputData.aspx

4.7.1.2 Input Specifications

1. The user will be presented with a form showing all of the data the user enter.
    a. The user will be able to select any of the data cells in order to edit the data.
2. After the user has reviewed the data, they will be able to select the “Save” or “Cancel” button.
    a. If the “Save” button is clicked a message will appear confirming the data has been submitted to the project’s PI/Admin(s).
3. The user will be taken back to their dashboard homepage.

4.7.1.3 Output Specifications

1. If the “Save” button is clicked a message will appear confirming the data has been submitted to the project’s PI/Admin(s).
2. The user will be taken back to their dashboard homepage.

4.7.1.4 Intercase Dependencies

1. The user must login successfully and needs to be an active user.
2. The user must have permissions to access the specific project.
3. The user needs to have finished and clicked “Next” before the committing data.

4.8 Feature 5: Add/Remove/Edit User Interface

4.8.1 Test Case 1: Add User

1. The user must click the “Manage Users” tab at the top of the STORC Dashboard Webpage
2. The user will be taken to the “Manage Users” page and do following:
    a. Click the “Add User” button.
    b. Search for the new user “Name” or “Saclink ID” in the search textbox.
    c. Select the appropriate user information a selection box.
    d. Click the “Add User” button at the bottom of the screen.

4.8.1.1 Test Items

Feature	Description	Pages
Add/Remove/Edit Users Interface	Add or Remove Permissions or Access - A series of drop-down boxes and buttons could add or remove access to each user on the edit list. These changes would be shown in the list on the right for clarity.	ManageUsers.html ManageUsers.cs ManageUsers.js ManageUsers.aspx

4.8.1.2 Input Specifications

1. Click the “Add User” button.
2. Search for the new user “Name” or “Saclink ID” in the search textbox.
3. Select the appropriate user information a selection box.
4. Click the “Add User” button at the bottom of the screen.

4.8.1.3 Output Specifications

1. A window will appear letting the PI/Admin know that the selected user/s is added to the edit list.
2. The users will also be added to the list of all users.

4.8.1.4 Intercase Dependencies

1. The PI/Admin must login successfully and needs to be an active user.
2. The PI/Admin must have permissions to manage users.
3. The user being added must have an active “SacLink” account.

4.8.2 Test Case 2: Remove User

1. The user must click the “Manage Users” tab at the top of the STORC Dashboard Webpage
2. The user will be taken to the “Manage Users” page and do following:
   a. Click the “Remove User” button.
   b. Search for the new user “Name” or “Saclink ID” in the search textbox.
   c. Select the appropriate user information a selection box.
   d. Click the “Remove User” button at the bottom of the screen.

4.8.2.1 Test Items

Feature	Description	Pages
Add/Remove/Edit Users Interface	Add or Remove Permissions or Access - A series of drop-down boxes and buttons could add or remove access to each user on the edit list. These changes would be shown in the list on the right for clarity.	ManageUsers.html ManageUsers.cs ManageUsers.js ManageUsers.aspx

4.8.2.2 Input Specifications

1. Click the “Remove User” button.
2. Search for the new user “Name” or “Saclink ID” in the search textbox.
3. Select the appropriate user information a selection box.
4. Click the “Remove User” button at the bottom of the screen.

4.8.2.3 Output Specifications

1. A window will appear letting the PI/Admin know that the selected user/s is removed to the edit list.
2. The users will also be removed to the list of all users.

4.8.2.4 Intercase Dependencies

1. The PI/Admin must login successfully and needs to be an active user.
2. The PI/Admin must have permissions to manage users.
3. The user being added must have an active “SacLink” account.

4.8.3 Test Case 3: Edit User Permissions

1. The user must click the “Manage Users” tab at the top of the STORC Dashboard Webpage
2. The user will be taken to the “Manage Users” page and do following:
   a. Click the “Edit User” button.
   b. Search for the new user “Name” or “Saclink ID” in the search textbox.
   c. Select the appropriate user information a selection box.
   d. Change appropriate permissions for that user.
   e. Click the “Confirm Change” button at the bottom of the screen.

4.8.3.1 Test Items

Feature	Description	Pages
Add/Remove/Edit Users Interface	Select between Add User, Delete User, Edit Permissions - A list of all current users could be filtered and added to an edit list. All users on the edit list would then be affected by any changes the admin or PI selects.	ManageUsers.html ManageUsers.cs ManageUsers.js ManageUsers.aspx

4.8.3.2 Input Specifications

1. Click the “Edit User” button.
2. Search for the new user “Name” or “Saclink ID” in the search textbox.
3. Select the appropriate user information a selection box.
4. Change appropriate permissions for that user.
5. Click the “Confirm Change” button at the bottom of the screen.

4.8.3.3 Output Specifications

1. A window will appear letting the PI/Admin know that the selected user’s account has been edited.
2. The application will also list the roles of all users profiles.

4.8.3.4 Intercase Dependencies

1. The PI/Admin must login successfully and needs to be an active user.
2. The PI/Admin must have permissions to manage users.

4.9 Feature 6: View and Select Pending Data Interface

4.9.1 Test Case 1: View Pending Data

1. The PI/Admin will click on the "Pending Submissions" tab at the top of the Dashboard
2. The PI/Admin will be taken to the "Pending Submissions" page
3. The PI/Admin will click on the "Approve Submissions" button
4. A list of Users and their pending entries will be presented to the PI/Admin for selection

4.9.1.1 Test Case

Feature	Description	Pages
View/Select Pending Data	View and Select Pending Data - By clicking on a particular user submission, the data for that submission could be viewed beneath in a separate window. Once this data has been checked for errors and committed, a check mark could indicate that this particular technician's submission has been completed. The Admin or PI could then move on to the next item.	InputData.html, InputData.cs, InputData.js, InputData.aspx

4.9.1.2 Input Specifications

1. The PI/Admin will click on the "Pending Submissions" tab at the top of the Dashboard
2. The PI/Admin will be taken to the "Pending Submissions" page
3. The PI/Admin will click on the "Approve Submissions" button
4. A list of Users and their pending entries will be presented to the PI/Admin for selection

4.9.1.3 Output Specifications

1. A window will appear showing the pending Project Data.

4.9.1.4 Intercase Dependencies

1. The PI/Admin must login successfully and needs to be an active user.
2. The PI/Admin must have permissions to view pending data.
3. There must be data in the pending queue.

4.9.2 Test Case 2: Edit Pending Data

1. From the Manage Data page the PI/Admin will click on the "Pending Submissions" tab
2. The PI/Admin will be taken to the "Pending Submissions" page

3. A list of Users and their pending entries will be presented to the PI/Admin for selection.
4. The PI/Admin will select one of the pending data entries.
5. PI/Admin can right-click the “Commit” , “Edit”. or “Reject” button
6. Once a choice has been chosen the data will then be removed from the pending submissions
    a. If committed the data will be appended into the database for the project
    b. If rejected the data will be deleted from all databases
    c. If edited the PI/Admin will edit the fields that need to be changed
        i. Once PI/Admin confirms the change the data will be appended into the database for that specific project

4.9.2.1 Test Case

Feature	Description	Pages
View/Select Pending Data	View and Select Pending Data - By clicking on a particular user submission, the data for that submission could be viewed beneath in a separate window. Once this data has been checked for errors and committed, a check mark could indicate that this particular technician’s submission has been completed. The Admin or PI could then move on to the next item.	InputData.html, InputData.cs, InputData.js, InputData.aspx

4.9.2.2 Input Specifications

1. From the Manage Data page the PI/Admin will click on the “Pending Submissions” tab
2. The PI/Admin will be taken to the “Pending Submissions” page
3. A list of Users and their pending entries will be presented to the PI/Admin for selection.
4. The PI/Admin will select one of the pending data entries.
5. PI/Admin can right-click the “Commit”, “Edit”. or “Reject” button
    a. Commit the pending data by clicking on “Commit”
    b. Reject the pending data by clicking on “Reject”
    c. Edit the pending data by clicking on “Edit”
        i. This will open up a web form with all fields filled out according to the pending data.
        ii. Once all data has been changed click “Commit”

4.9.2.3 Output Specifications

1. The output is as follows:
    a. “Commit”
        i. An message will appear confirming the data has been successfully committed.
        ii. This data will be reflected in the database

    b. “Reject”
        i. An message will appear confirming the data has been successfully deleted.
    c. “Edit”
        i. An message will appear confirming the data has been successfully committed.
        ii. This data will be reflected in the database

4.9.2.4 Intercase Dependencies

1. The PI/Admin must login successfully and needs to be an active user.
2. The PI/Admin must have permissions to view pending data.
3. There must be data in the pending queue.

5. SYSTEM TEST / REQUIREMENTS TRACEABILITY

5.1 System Test / Requirements Specification / Design Component Traceability Matrix

System Test	Requirement Specification	Design Component
4.4 Feature 1: Homepage Management Interface	2.3.1 UC-4: Customize Dashboard	3.7 Select/Edit Project or Station Interface
4.4.1 Test Case 1: Customizing Widget	2.3.1 UC-4: Customize Dashboard	3.7 Select/Edit Project or Station Interface
4.4.2 Test Case 2: Delete Widget	2.3.1 UC-4: Customize Dashboard	3.7 Select/Edit Project or Station Interface
4.4.3 Test Case 3: Adding Widget	2.3.1 UC-4: Customize Dashboard	3.7 Select/Edit Project or Station Interface
4.4.4 Test Case 4: Move/Resize Widget	2.3.1 UC-4: Customize Dashboard	3.7 Select/Edit Project or Station Interface
4.4.5 Test Case 5: Create New Project	2.3.1 UC-4: Customize Dashboard	3.7 Select/Edit Project or Station Interface
4.5 Feature 2: Project Selection Interface	2.3.1 UC-4: Customize Dashboard	3.2 Project Selection Interface
4.5.1 Test Case 1: Select Project and Station	2.3.1 UC-4: Customize Dashboard	3.2 Project Selection Interface
4.6 Feature 3: Data Input Interface	2.3.2 UC-3: Input Data	3.3 Data Input Interface
4.6.1 Test Case 1: Input Data	2.3.2 UC-3: Input Data	3.3 Data Input Interface
4.7 Feature 4: View and Submit Data Interface	2.3.2 UC-3: Input Data	3.6 View/Select Pending Data
4.7.1 Test Case 1: Review/Submit Data	2.3.4 UC-3 Manage Data	3.4 Review and Submit Data Interface
4.8 Feature 5: Add/Remove/Select User Database	2.3.3 UC-2 Manage Users	3.5 Add/Remove/Edit User Interface
4.8.1 Test Case 1: Add User	2.3.3 UC-2 Manage Users	3.5 Add/Remove/Edit User Interface
4.8.2 Test Case 2: Remove User	2.3.3 UC-2 Manage Users	3.5 Add/Remove/Edit User Interface
4.8.3 Test Case 3: Edit User Permissions	2.3.3 UC-2 Manage Users	3.5 Add/Remove/Edit User Interface
4.9 Feature 6: View and Select Pending Data Interface	2.3.2 UC-3 Input Data	3.6 View/Select Pending Data
4.9.1 Test Case 1: View Pending Data	2.3.2 UC-3 Input Data	3.6 View/Select Pending Data
4.9.2 Test Case 2: Edit Pending Data	2.3.2 UC-3 Input Data	3.6 View/Select Pending Data

6. APPROVALS

By signing you agree that all conditions and commitments to the project are accurate to the best of your knowledge. I certify that the information in this System Test Specification is correct and the senior project group *CD Jam* can continue on with the design of the project. I also certify that I will follow and provide all needing requirements stated in this document and that I am willing to follow through with all conditions.

CD Jam Team members:

X ______________________

Cole Culler

X ______________________

David Grapentine
Project Lead

X ______________________

Ashley Gregory

X ______________________

John Jones

X ______________________

Michael Smith

Faculty Advisor:

X ______________________

Ying Jin
Faculty Advisor

APPENDIX A – SOFTWARE PROBLEM REPORT TEMPLATE

This section provides a sample template for reporting software problems that are discovered during the course of performing the test cases detailed in this document. The template is located on the next page.

SOFTWARE PROBLEM REPORT FOR STORC DASHBOARD

Problem Report ID #: _____

Program Part: ______________________ **Release: ________** **Version: ____**

Report Type:

□	Coding Error	□	Documentation
□	Design Error	□	Hardware
□	Suggestion	□	Query

Severity:

□	Fatal
□	Serious
□	Minor

Attachments (List attachments): ____

Problem Summary: ____

Can you reproduce the problem? □ Yes □ No

Problem and how to reproduce it: ____

____

____

____

____

Suggested Fix (optional): ____

____

____

____

____

Reported By: _________________ **Date: ___/___/2015**

Items Below Are For Use Only By the Development Team

Functional Area:______________________ **Assigned To: ________________________**

Comments: ____

____

____

____

Status:

□	Open	□	Closed

Priority:

□	High	□	Medium	□	Low

Resolution: **Resolution Version NO: _________**

□	Pending	□	Deferred	□	Withdrawn by reporter
□	Fixed	□	As designed	□	Need more info
□	Irreproducible	□	Can’t be fixed	□	Disagree with suggestion

Resolved By: _________________ **Date: ___/___/2015**

Resolution Tested By: _________________ **Date: ___/___/2015**

Treat as Deferred: □ Yes □ No